

IMPLEMENTASI FORWARD DYNAMIC PROGRAMMING DALAM MENENTUKAN PORTOFOLIO INVESTASI

Tiwuk Widiastuti

Jurusan Ilmu Komputer, Fakultas Sains dan Teknik Universitas Nusa Cendana

ABSTRACT

A company usually has to make a decision from variety of investment proposal. The challenge is how to get the right option in order to get the highest benefit. Most of them consist of types of investment, expenses and how much income would we get. Certainly, cost and income are the most important thing. Based on the given problem I would like to assembly a software to determine a investment portfolio from some investment proposal. The goal is the best result computed by computer through that software, consider getting the highest advantage but spending the lowest cost. Forward Dynamic Programming is a method in The Operation Research which is suitable to solve the investment problem. The method would calculated variables through stages, how many stages would depend on number of the proposal and how many types of investment's cost and income. The result will be the best decision implemented by a company.

Keywords: *Investment Portfolio, Forward Dynamic Programming*

PENDAHULUAN

Masalah pengoptimalan sudah menjadi masalah yang lazim dijumpai dimana-mana. Dari lingkungan yang terkecil, yaitu keluarga sampai pada lingkungan perusahaan besar. Umumnya yang dititikberatkan adalah pengoptimalan di bidang finansial, yaitu memaksimalkan pendapatan atau meminimalkan pengeluaran atau dengan

kata lain memaksimalkan laba. Sebagai contoh, di lingkungan perusahaan pengangkutan barang atau perusahaan *cargo*, pengiriman barang diatur sedemikian rupa sehingga tidak terjadi pemborosan waktu, tempat dan biaya. Tentunya pengaturan dilakukan dengan menggunakan rumus-rumus matematis yang berkaitan dengan pengoptimalan,

sehingga diperoleh suatu hasil penghitungan yang tepat.

Selain optimalisasi pada pengangkutan barang masih terdapat banyak lagi bidang-bidang lain yang melakukan hal yang sama. Masalah alokasi pembagian suatu modal yang tersedia adalah salah satu kasus yang sering dihadapi oleh perusahaan-perusahaan. Alokasi modal ini berkaitan dengan usaha memperoleh keuntungan maksimal dengan memperhitungkan biaya-biaya pengeluaran. Dari beberapa alternatif harus dipilih salah satu keputusan yang terbaik. Pilihan terbaik adalah pilihan dengan total pengeluaran terkecil dan total laba terbesar yang akan diperoleh. Untuk memperoleh suatu kemajuan finansial maka pilihan-pilihan terbaiklah yang harus dicari. Jika terjadi kesalahan penghitungan dalam mencari pilihan terbaik maka laba maksimal yang diharapkan tidak akan diperoleh. Bahkan pengeluaran yang lebih besar dari laba yang diperoleh bisa saja terjadi. Jika hal ini terjadi pada suatu perusahaan maka kerugianlah yang akan ditanggung.

Salah satu faktor yang mengakibatkan terjadinya kesalahan penghitungan adalah jumlah data yang sangat besar. Semakin banyak data-data yang dihitung, semakin tinggi pula tingkat kerumitannya. Untuk menghitung data-data dalam bentuk angka, penghitungan secara manual telah

lama ditinggalkan, terlebih untuk jumlah data yang relatif banyak. Penghitungan data-data telah diganti dengan menggunakan alat-alat bantu elektronik, misalnya kalkulator dan komputer. Dengan alat-alat bantu tersebut kesalahan-kesalahan yang bersifat manusiawi dapat dikurangi.

MATERI DAN METODE

Pengertian Dynamic Programming.

Umumnya penyelesaian suatu masalah dalam metode Riset Operasi menghasilkan suatu keputusan tunggal. Biasanya keputusannya adalah hasil proses optimalisasi. Variabel-variabel awal dimasukan secara serentak dan dalam penghitungan terjadi pengulangan sampai memperoleh kondisi akhir yang ditentukan. Berakhirnya pengulangan penghitungan akan menghasilkan suatu nilai sebagai jawaban optimal. Pada masalah lain, terdapat kemungkinan bahwa masalah dapat dipecah-pecahkan menjadi beberapa bagian. Bagian-bagian kecil dihitung terlebih dahulu untuk memperoleh keputusan-keputusan. Penggabungan keputusan-keputusan adalah hasil akhir sebagai jawaban yang diinginkan.

Pada tahun 1950-an, Richard Bellman mengembangkan suatu metode yang berguna dalam optimalisasi. Konsep dasar dari metode tersebut adalah

optimalisasi dari beberapa variabel yang dibagi dalam beberapa tahap. Optimalisasi dilakukan seiring dengan penyelesaian tahap demi tahap. Richard Bellman menyebutnya dengan *Dynamic Programming*.

Dalam Riset Operasi metode ini merupakan suatu metode yang cukup handal untuk menyelesaikan persoalan optimalisasi yang dilakukan secara bertahap. *Dynamic Programming* adalah suatu teknik matematis yang digunakan untuk membuat suatu keputusan dari serangkaian keputusan yang saling berkaitan. Metode ini dirancang menyederhanakan suatu penghitungan sehingga akan lebih mudah dalam penerapannya. Teori utama dalam *Dynamic Programming* adalah prinsip optimalitas. Prinsip ini menentukan bagaimana menguraikan atau memecahkan masalah dan menjawabnya dengan benar. Tidak ada formulasi matematis yang standar. Setiap masalah akan mempunyai struktur penyelesaian yang berbeda.

Metodologi

Secara umum Dynamic Programming digunakan untuk menyelesaikan masalah yang melibatkan berbagai keputusan yang saling berhubungan. Ada beberapa istilah yang digunakan sebagai berikut:

a. **Stage.** Langkah pertama harus dilakukan dalam memformulasikan

Dynamic Programming adalah mengidentifikasi *stage* atau tahapan dalam proses keputusan. Misalnya, terdapat n tahapan yang diberi label $1, 2, 3, \dots, n-1, n$.

b. **States.** Diidentifikasi berdasarkan setiap *stage* yang menggambarkan status semua informasi dalam membuat keputusan. Misalnya jika dalam *stage* 3 terdapat *state* 5, maka ditulis dengan $s_3 = 5$.

c. **Decisions.** Variabel keputusan atas *state* i ditulis dengan ' d_i '. Misalnya dalam *stage* 3 terdapat *state* 5, maka d_3 mungkin sama dengan 7 atau 8 (menuju *state* 7 atau 8).

d. **Return Function.** Untuk menghitung efektifitas digunakan fungsi dengan notasi f yang dapat berupa biaya, laba, jarak atau beberapa hitungan yang lain. Fungsi f tersebut dinamakan **Return Function**, misalnya $f_i(s_i, d_i)$. Apabila nilai f_i optimum ditulis dengan $f_i^*(s_i)$ atau dikenal dengan istilah *optimum return function*. Misalnya *stage* 2 terdapat *states* 2 atau $i=2$, dan $s_i=2$, maka fungsinya ditulis :

$$F_2 = \min f_2(2, d_2)$$

$$d_2 = \min \{f_2(2,4), f_2(2,5), f_2(2,6)\}$$

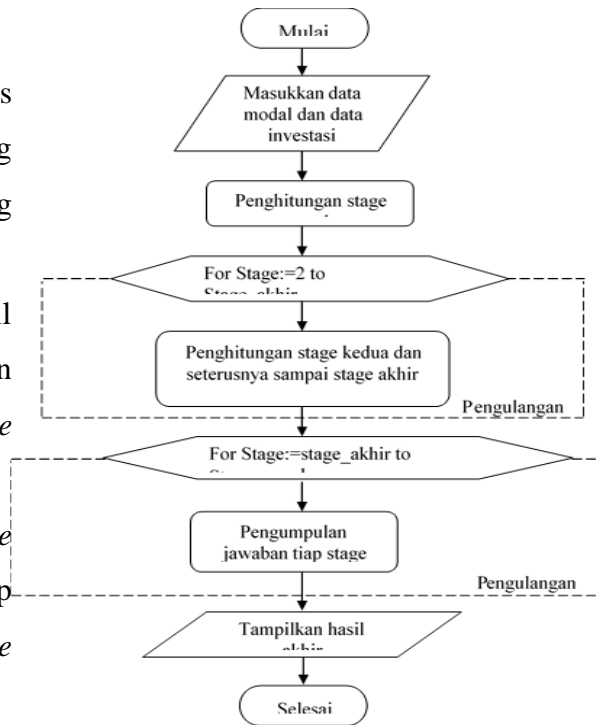
Walaupun tidak ada formula matematis yang standar dalam *Dynamic Programming* tapi ada ciri-ciri pokok yang menandai metode ini. Beberapa

gambaran dasar yang menandai persoalan *Dynamic Programming* adalah:

- a. Persoalan dapat dibagi menjadi beberapa *stage*, yang pada masing-masing *stage* diperlukan adanya suatu keputusan.
- b. Masing-masing *stage* terdiri atas sejumlah *state* yang saling berhubungan dengan *stage* yang bersangkutan
- c. Hasil dari keputusan yang diambil pada setiap *stage* ditransformasikan dari *state* yang bersangkutan ke *state* yang berikutnya pula.
- d. Keputusan terbaik pada suatu *stage* bersifat *independent* terhadap keputusan yang dilakukan *stage* sebelumnya.
- e. Prosedur pemecahan persoalan dimulai dengan mendapatkan keputusan terbaik untuk setiap *state* dari *stage* terakhir.
- f. Ada suatu hubungan timbal-balik yang mengidentifikasi keputusan terbaik untuk setiap *state* pada *stage* n , berdasarkan keputusan terbaik untuk setiap *state* pada *stage* $(n+1)$.
- g. Dengan menggunakan hubungan timbal-balik ini, prosedur penyelesaian persoalan bergerak mundur *stage* demi *stage*, pada setiap *stage* berusaha diperoleh keputusan optimum untuk masing-masing *state* hingga akhirnya diperoleh keputusan

optimum yang menyeluruh mulai dari *stage* awal.

Urutan logika program secara umum digambarkan pada flowchart sebagai berikut:



Gambar 1. Flowchart Umum Program

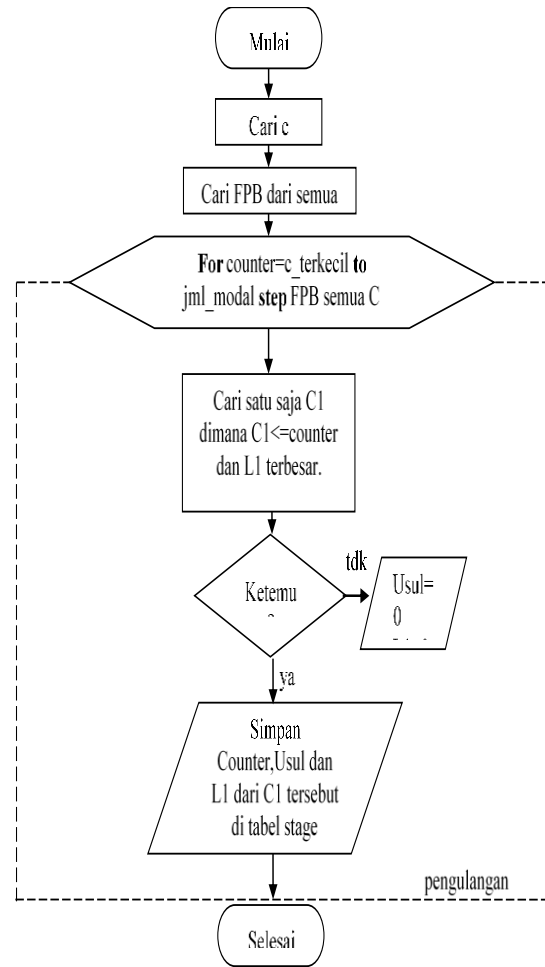
Penghitungan Stage I

Proses penghitungan akan melalui beberapa tahap atau *stage*. Proses dimulai dari tahap pertama sampai pada tahap terakhir dari suatu kasus. Setiap kasus mempunyai jumlah tahap perhitungan yang berbeda beda. Sesuai dengan metode *Dynamic Programming* untuk kasus ini, tahap pertama mempunyai cara penghitungan yang sedikit berbeda dengan tahap kedua dan selanjutnya, karena tahap pertama tidak mempunyai tahap sebelumnya untuk dipertimbangkan.

Sedangkan tahap kedua dan seterusnya akan mempunyai tahap sebelumnya untuk dipertimbangkan..

Setiap hasil penghitungan dimasukkan ke dalam tabel Stage. Pada proses penghitungan tabel ini akan berfungsi sebagai acuan untuk menghitung tahap demi tahap. Setelah proses perhitungan selesai, tabel ini akan berfungsi untuk pencarian hasil akhir yang optimal.

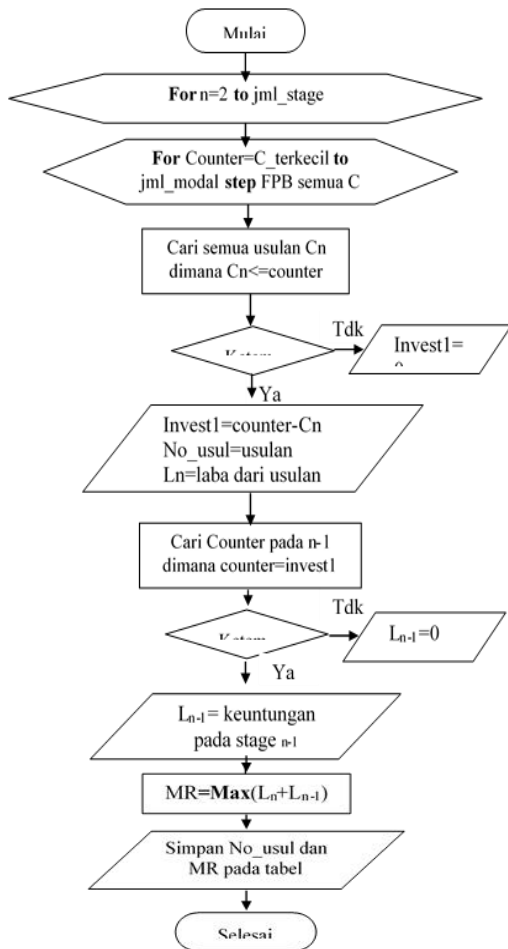
Penghitungan tahap pertama adalah penghitungan yang dilakukan pada investasi atau C dan Laba atau L untuk pihak pertama. Prosesnya mengalami perulangan sampai pada *state* akhir ditandai dengan jumlah *counter* yang mencapai nilai tertentu. *Counter* akan berhenti jika nilainya sama dengan jumlah modal. *Counter* adalah nilai-nilai hasil penguraian modal dengan penambahan dengan nilai tertentu yang tetap. Pada setiap penambahan *counter* akan dicari usulan mana yang sesuai dari pihak pertama untuk dimasukkan ke dalam tabel stage. Usulan yang sesuai adalah usulan dengan hitungan investasi atau C yang tidak melebihi *counter*. Jika terdapat lebih dari satu usulan akan diambil usulan dengan laba terbesar. Proses penghitungan tahap pertama dapat digambarkan dalam flowchart berikut:



Gambar 2. Flowchart Stage I

Penghitungan Tahap II dan Seterusnya

Flowchart berikut akan menggambarkan proses penghitungan tahap II dan seterusnya:



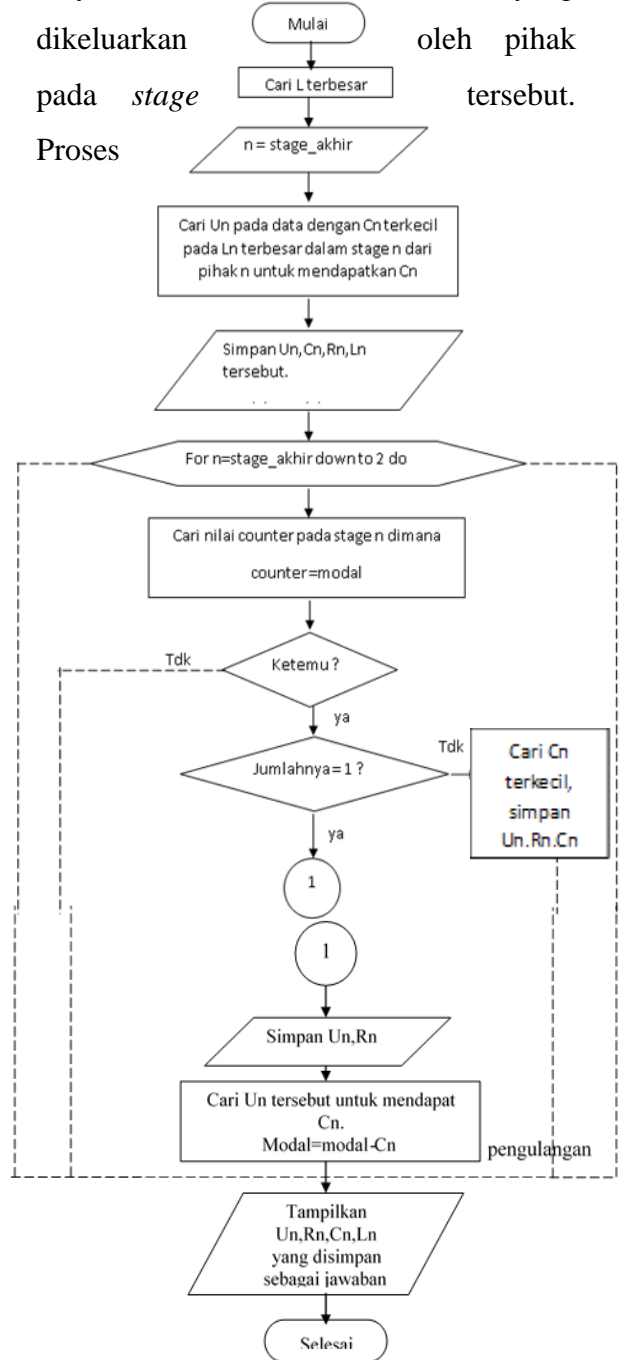
Gambar 3. Flowchart Penghitungan II dan seterusnya

2.2.3 Proses Pencarian Hasil Akhir

Semua penghitungan dari stage pertama sampai pada stage akhir disimpan dalam suatu tabel Stage. Setelah penghitungan selesai tabel tersebut berfungsi untuk mencari hasil akhir.

Pencarian hasil akhir dimulai dari stage dengan total keuntungan terbesar. Kemudian pencarian dilakukan secara mundur sampai pada stage pertama. Setiap perpindahan stage, pencarian didasarkan pada jumlah modal yang tersedia. Modal akan berkurang sesuai dengan usulan

biaya investasi awal yang dikeluarkan oleh pihak pada stage tersebut. Proses



pencarian hasil akhir digambarkan pada flowchart berikut:

Hasil Penghitungan

Modal : Rp 12.000.000.000 Periode 01/01/2003 s/d 01/01/2004

Pihak	Usulan	Jenis Usaha	Investasi	Pendapatan	Laba
1	4	Toko Komputer	Rp 7.500.000.000	Rp 24.000.000.000	Rp 16.500.000.000
2	1	Waral	Rp 1.000.000.000	Rp 2.400.000.000	Rp 0
3	5	Penyewaan VCD	Rp 1.000.000.000	Rp 2.500.000.000	Rp 1.500.000.000
4	1	Salon	Rp 2.000.000.000	Rp 4.000.000.000	Rp 2.000.000.000

Sisa Modal Rp 500.000.000
 Total Investasi Rp 11.500.000.000
 Total Pendapatan Rp 20.500.000.000
 Total Laba Rp 33.400.000.000

Gambar 4. Flowchart Pencarian Hasil Akhir

HASIL DAN PEMBAHASAN

Program yang dihasilkan akan menangani masukan data dari pemakai program, program akan mengolahnya dengan metode *Dynamic Programming* dan menghasilkan keluaran sebagai jawaban akhir.

Program terdiri dari beberapa form dengan satu form utama sebagai form pengontrol untuk mengaktifkan form-form lain. Form utama ini juga sebagai form bagi pemakai program untuk memasukkan data dan melihat hasil penghitungan. Form-form lain adalah form-form yang bersifat membantu pemakai program dalam menggunakan program ini.

Form Utama

Ketika program dijalankan, form yang akan ditampilkan adalah form utama. Kondisi awal form ini adalah form tanpa berisi data. Form dengan keadaan tanpa data tentunya belum dapat difungsikan. Agar dapat digunakan maka pemakai

aplikasi harus terlebih dahulu mengisi data-data yang diperlukan atau menggunakan fasilitas yang ada untuk membuka data yang pernah disimpan.

Data-data yang dibutuhkan adalah jumlah modal, periode investasi, jenis usaha, investasi awal, pendapatan dan laba. Setelah data-data tersebut diisi dengan benar maka proses penghitungan dapat dilakukan oleh komputer dengan menekan tombol Hitung.

Form Utama dengan data-data awal terlihat seperti pada gambar 5 :

Dynamic Programming - Penentuan Portofolio Investasi

Berkas: Berjalan, Lain-lain

Baru, Buka, Simpan, Cetak Hasil, Hitung, Print Help

Nama File: C:\My Documents\DATA\sample5.db Periode: 01/01/2003 s/d 01/01/2004

Data usulan investasi

Modal: Rp 12.000.000.000

Usulan	Jenis Usaha 1	Investasi 1	Pendapatan 1	Laba 1	Jenis
1	Toko Buku	Rp 5.000.000.000	Rp 10.000.000.000	Rp 5.000.000.000	Waral
2	Toko Roti	Rp 5.500.000.000	Rp 10.000.000.000	Rp 4.500.000.000	Waral
3	Toko Obat	Rp 6.500.000.000	Rp 12.500.000.000	Rp 6.000.000.000	Rent.
4	Toko Komputer	Rp 7.500.000.000	Rp 24.000.000.000	Rp 16.500.000.000	Ganw.
5	Toko Besi	Rp 12.000.000.000	Rp 48.000.000.000	Rp 12.000.000.000	Rent.

Sisa Data

Dempun modal: 0%

maka pihak dan usulan yang dipilih dengan laba optimal adalah:

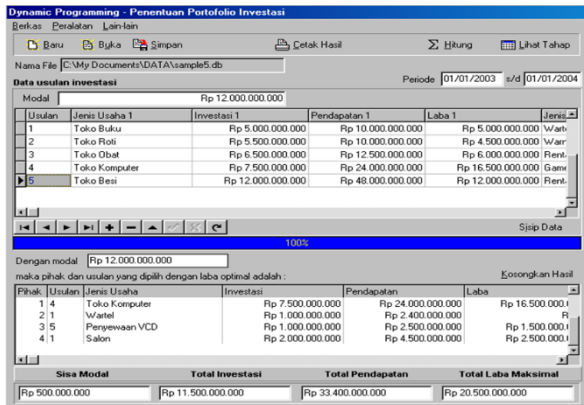
Pihak	Usulan	Jenis Usaha	Investasi	Pendapatan	Laba

Sisa Modal Total Investasi Total Pendapatan Total Laba Maksimal

Gambar 5. Form Utama

Setelah prosedur di atas di kerjakan maka form utama akan menampilkan data-data hasil penghitungan. Keluaran yang dihasilkan pada grid bagian keluaran dari form utama ini adalah usulan-usulan investasi yang dipilih dengan pertimbangan memiliki total laba terbesar dimana investasi yang akan dibiayai tidak melebihi modal yang

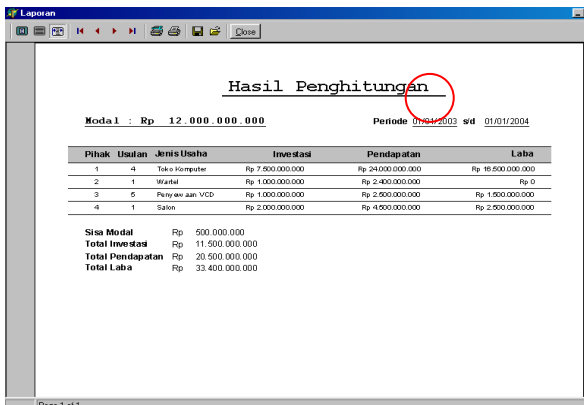
tersedia. Hasilnya seperti terlihat pada gambar 6 :



Gambar 6. Form utama dengan hasil perhitungan

Form Laporan

Form ini berfungsi untuk menampilkan hasil cetakan laporan di layar monitor sebelum laporan hasil perhitungan dicetak di printer. Bentuk laporan pada form Laporan dapat dilihat pada gambar 7 :



Gambar 7. Form Laporan

SIMPULAN

Beberapa kesimpulan yang dapat diambil dari hasil implementasi *Dynamic Programming* dalam mengoptimalkan anggaran modal adalah :

1. Pada kasus penentuan portofolio investasi, metode *Dynamic Programming* hanya dapat memberikan satu jawaban yang optimal dengan total pendapatan terbesar, dimana total pengeluaran juga dipertimbangkan yang mengarah pada usaha memperoleh laba terbesar.
2. Dengan variasi variabel-variabel masukan program tetap menghasilkan jawaban yang optimal. Jawaban yang optimal berupa total laba terbesar bersama usulan-usulan investasi yang dipilih dengan total biaya investasi yang tidak melebihi modal.
3. Semakin banyak jumlah *state* dan *stage* maka semakin lama pula proses penghitungannya. Jumlah *state* pada tiap *stage* bergantung pada besarnya perbedaan data-data biaya investasi yang dihitung. Jumlah *stage* bergantung pada jumlah pihak yang mengajukan usulan investasi.
4. Jawaban akhir yang optimal hanya dipengaruhi biaya untuk investasi dan laba tapi tidak dipengaruhi oleh jenis usaha.

Berdasarkan hasil implementasi dan analisis yang telah dilakukan maka penulis mengusulkan beberapa saran yang berguna bagi pembaca dan pemakai program, saran-saran yang dapat diberikan adalah sebagai berikut :

1. Program aplikasi ini bukan satu-satunya alat untuk dapat memberi keputusan optimal dalam memilih usulan-usulan investasi terbaik. Untuk itu diperlukan pertimbangan lain untuk memperoleh usulan investasi terbaik.
2. Program aplikasi ini dapat dikembangkan sehingga dapat menangani masalah pengoptimalan anggaran modal dengan memperhatikan rincian investasi, biaya-biaya dan rincian pendapatan.

DAFTAR PUSTAKA

- Dimiyati, Tjuju Tarlian – Ahmad, 1994. *Operations Research*, Sinar Baru Algesindo Offset, Bandung.
- Jayanto, 2000. *Membuat Aplikasi Database dengan Delphi*, PT. Elex Media Komputindo, Jakarta.
- Pranata, Antony, 2000. *Pemrograman Borland Delphi*, Edisi 3, Andi Offset, Yogyakarta.
- Taha, Hamdy A, 1996. *Riset Operasi Suatu Pengantar*, Jilid 1, Binarupa Aksara, Jakarta.
- Zulian, Yamit, 1993. *Manajemen Kuantitatif Untuk Bisnis*, Edisi 1, BPFE, Yogyakarta